# Design of the Security Module for Safe Data Sending in a Web System

*Seung-Ju Jang*

Dong-Eui University, Dept. of Computer Engineering

**Summary**
According to increasing use of internet, the security of network connection between web server and client is equal concern to both parties. Such risks of network security are eavesdropping, fraud, interception and so on. Therefore it is important to protect data from/to internet data. In special, there are two types of internet security, one is network security, the other is data security. Data security is main concern in web data. In this paper, I use Apache Web Server and LSP(Layered Service Provider) client protocol. I designed secure server and client module. Both modules have RSA encryption algorithm based on PKI(Public Key Infrastructure) and SHA-1 hash algorithm to provide data integrity.

***Key words:***
*Security Module, Web Security, LSP protocol, Server Security, Client Security*

## 1. Introduction

The web server and client environments are available a hypermedia reference system that crawl WWW(World Wide Web). HyperMedia is generally unity one of HyperText and Multimedia. If one makes use of WWW, he/she can dynamically refer to information - text, graphic, image and sound - that is dispersed in all parts of the world [1, 3 5].

History of WWW had been started in CERN(Conseil Europeen pour la Recherche Nucleaire) which is a physics institute in Europe in March, 1989. The project by Tim Berners Lee which developed effective use of dispersed variety information is the beginning. Later another project had been studied and developed hypertext text processing technology in late 1990. The result had been introduced in HyperText '91 conference and had been announced as WWW in 1992. The first WWW had used telnet for connection over system. Therefore it was not worthy of notice on account of GOPHER information reference service with menu form at that time. The WWW had attracted public attention in announcing the development of MOSAIC which was a user interface to serve WWW by Mike Anderson at NCSA(National Center for Supercomputing Applications) in the U.S.A in June 1993. Thereafter Mart Anderson had founded Netscape Co. with Jim Clark in 1994. Netscape Co. announced a web browser 'Netscape' and made a conquest of web browser market along with Internet Explorer of Microsoft Inc[2, 4, 6].

Manuscript received August 5, 2007
Manuscript revised August 20, 2007

The main components of WWW are HTML(Hyper Text Markup Language) which is a standard language to draw up a hypertext document, HTTP(Hyper Text Transfer Protocol) which is a network protocol to transfer HTML document to/from a client and a server, and URL(Uniform Resource Locator) which is web address to indicate web server system. In addition there is an external viewer to represent sound, image, and moving pictures in the client side. Web action model of client and server is Fig. 1.

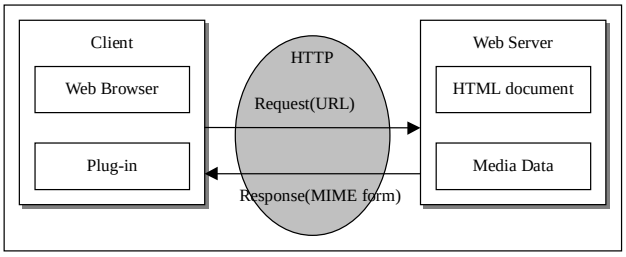


Fig. 1 web action model

In this paper I designed web system architecture in the secure data transmission. The web security environment has relationship both client and server. Security module of client is programmed with LSP layer among Windows socket programming environment. The Linux Operating System is available into security module of server. Apache web server is used to implement secure web server. Secure module environment was constructed into Apache web server [10].

This paper was composed of system architecture in chapter 2, secure web server/client in chapter 3, conclusion in chapter 4.

## 2. System Architecture

### 2.1 Apache Web Server Module

Apache server was born based on NCSA HTTPD 1.3 version which was most popularity in 1995. Afterwards the Apache web server was announced more elevation function at the NCSA web server. Netcraft is finding Table 1 statistics result which is in the middle of using

computer web server software that is being connected at internet. When one see result [Table 1], we know use tendency of portion of Apache web server increasing constantly [7, 8, 9].

Table 1: Use Tendency of Apache Web Server

| Server | Dec97 | Percent | Jan98 | Percent | Change |
|---|---|---|---|---|---|
| Apache | 753341 | 44.79 | 827893 | 45.12 | 0.33 |
| IIS | 351755 | 20.91 | 381763 | 20.81 | -0.10 |
| Netscape-Enterprise | 88616 | 5.27 | 97317 | 5.03 | 0.03 |
| NCSA | 68097 | 4.05 | 69223 | 3.77 | -0.28 |

In Apache web server, we can append new function module into it. Present much modules are appended into Apache server like CGI, SSL module.

### 2.2 Winsock 2 LSP(Layered Service Provider)

Windows socket 2 was laying transport layer at OSI network class. Winsock 2 had been Ied to WOSA(Windows Open Services Architecture) components. WOSA offer common interface of between application and service. The previous program do not need to alter even if one addition become or whether one function be upgraded. API(Application Program Interface) and SPI(Service Provider Interface) had been provided for practical application programmer [11, 12, 13].

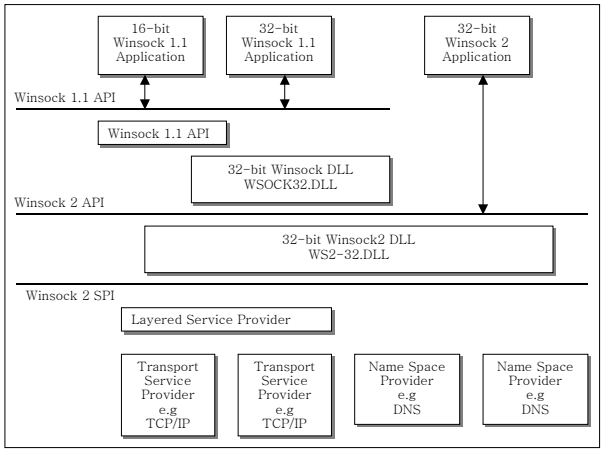


Fig. 2 Winsock 2 Architecture

Winsock Transport/Name Space Service Provider is DLL of entry point which is exported by initialized of service provider. Besides other Service Provider functions let approach Winsock 2 DLL through Dispatch Table within Service Provider. Service Provider DLL load to memory by Winsock 2 DLL at time necessity and unload memory at time unwanted.

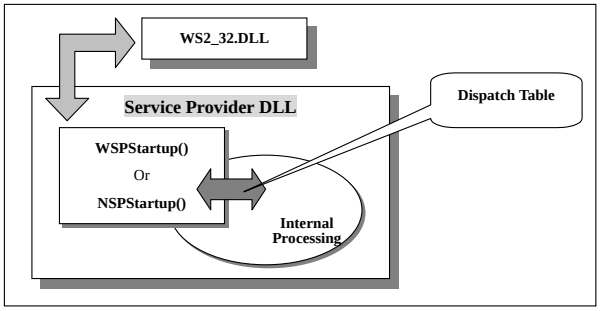


Fig. 3 Winsock SPI action model

## 3. Design of Secure Module in Web Server/Client Module

Action of security module in Apache web server is as following. For applying security module of server, we utilized C API(Application Program Interface) that offer from Apache web server. Abilities of server security module are applying HTML message to encrypted HTML message among with RSA encryption algorithm. RSA decryption algorithm and digital signature was used from CGI form input data. SHA-1 hash algorithm proves that message is not changed on the way of data transmission.

The client uses RSA decryption algorithm to decrypt for receiving encrypted message from server and confirms that message is not changed in the middle of data transmission. Security module had URL filtering ability to distinguish security web server data or not. URL filtering ability let security module fulfill when user request URL is security site. Otherwise, general processing module is managed. The input information in CGI form that is an enciphering information transfers to server.

Public key and private key length are being fixed to 512 bits that uses in RSA encryption algorithm at server and client. Action process of server and client is Fig 4.

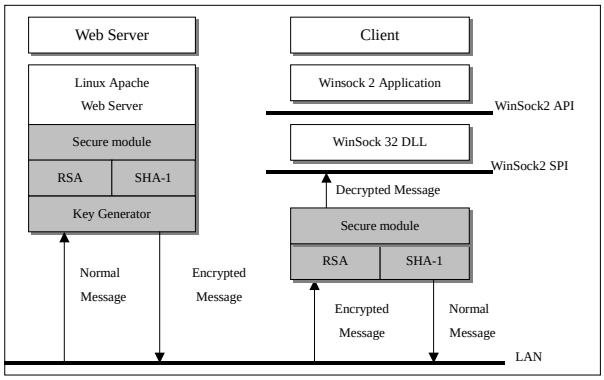


Fig. 4 Security Module of Server/Client Architecture

When a client requires a data from server, server encrypted web page contents for a client request using RSA encryption algorithm. Encrypted messages are sent to client. LSP secure module of client let encrypted message decrypt by RSA algorithm. encrypted message can be seen to web browser as normal message.

## 3.1 Design Secure Module of Apache Web Server

Apache web server send answer about client request. The request HTML file from client was seen to client again from server security module. In environment establishment of Apache web server if client request HTML file, the established secure server module (write_handler, mod_cgi_debug) is executed. Establishment environment file can be set whether regular module or security module. Security module had encrypting ability for HTML file and had decrypting ability for CGI response. The encrypting ability of security module is designed Digital signature algorithm and RSA algorithm which is used to encrypt for client request. RSA encryption algorithm reads as letter unit for the HTML data, attaches key value and let encryption message send to client. Encryption message is difficult to decrypt even if attacker intercept messages through network because of contents shape of a character of message enciphering.

Message integrity checking mechanism is as follow. Private/public key was used to encrypt message with RSA algorithm at server side and private/public key was used to decrypt messages at client side. In case messages are unable to decrypt, it indicates that the messages are changed during message transmission. If a client denies sending of messages, the fact will be proven that the message was sent by client as public/private key.

HTTP handling process of client is Fig. 5.

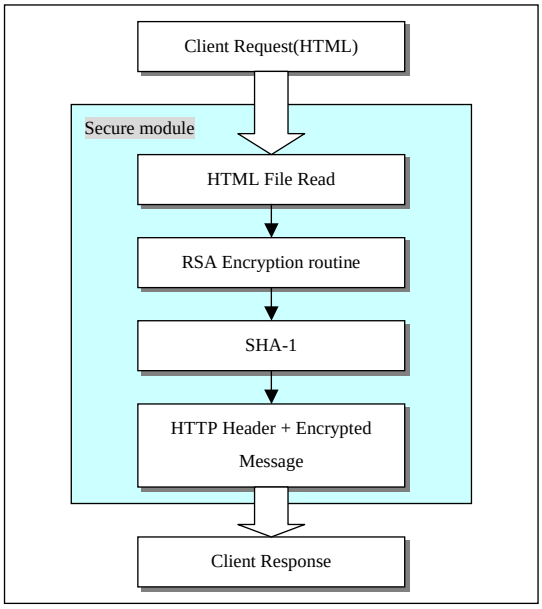


Fig. 5 HTTP Handling Process of Client

An answering message for CGI in client returns to encrypting message form. In the security module after encrypting message was decrypted, the CGI program is processed normally. Handling process of CGI in client is Fig. 6.

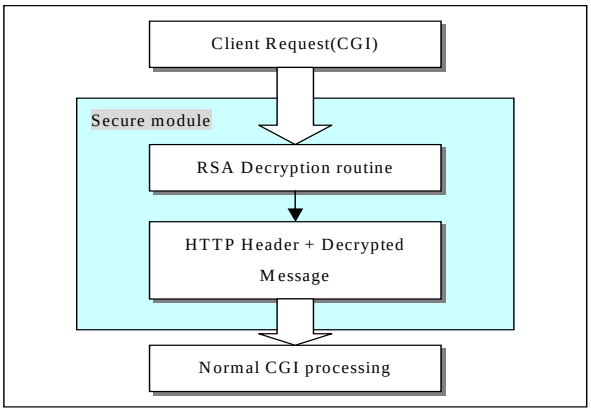


Fig. 6 Handling Process of CGI in client

## 3.2 Design Security Module of Client

Client demands HTML document of web server through web browser. In case of not installing secure module, encipher data was shown to web browser of client. Security module of client decrypts enciphered message from HTML document of server and normal message let to display at browser. Secure module of client utilized LSP functions which are lower SPI of upper level of API.

LSP utilized packet that enter from UDP(User Datagram Protocol). Organization of this packet is the HTTP header and an enciphered message. If changing HTTP header, the body data was not recognized from web browser. Therefore, HTTP header in the middle of packet should manage as normal procedure. Because the messages are enciphered by RSA algorithm at server, the enciphered messages should decrypt to normal messages. Security module of LSP decrypts enciphered message using private key of client. Decrypted messages display normal messages through API at browser. If decrypted messages are broken or displayed strange characters, it is proved that messages was changed during messages transmission. Procedure of secure module of client is Fig. 7.

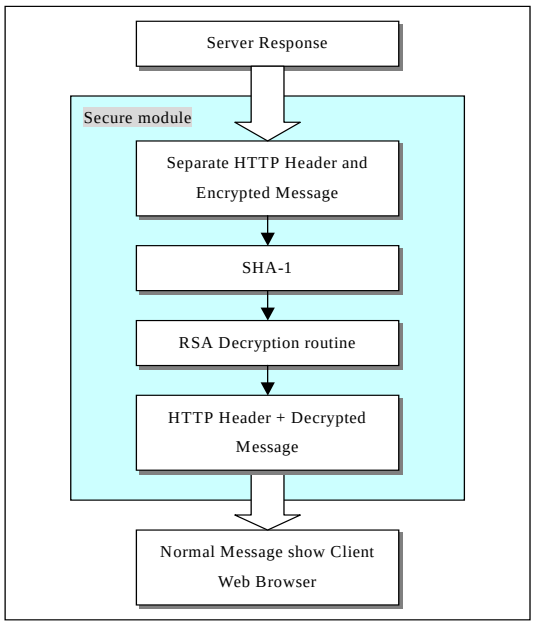


Fig. 7 Answering Process of Client

Security module of client is designed with connecting to security web server and general web server. When general web server data go through the security module of client, unexacting message appears at browser. For preventing such circumstances, URL web address was saved at configuration file. To differentiate security web server from general web server, the security module compares URL address that is input from web browser of client with registered URL address at file. Procedure of URL filtering in client is Fig. 8.

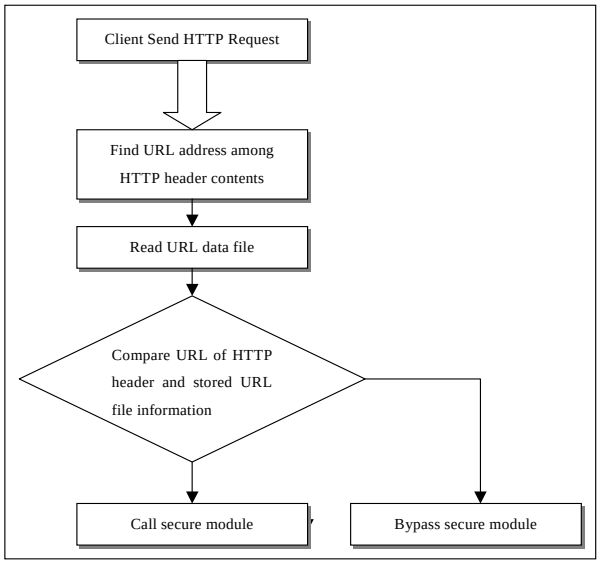


Fig. 8 Procedure of URL Filtering

The client is not supporting information from web server unilaterally but client send data to server. Such circumstances are in receiving of CGI form data, when one input data, press 'confirm' button and send it to the CGI interface of server. Before sending out network, encrypting input data of CGI form at client, and the encryption data is send-off to server through network. Procedure of encryption of input data of CGI form in client is Fig. 9.

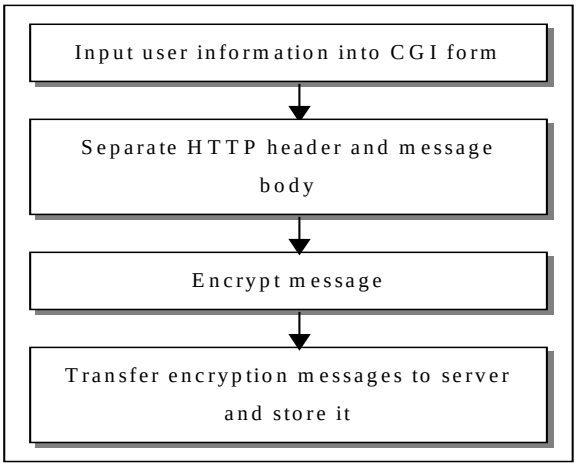


Fig. 9 Procedure of CGI request in server

## 4. Conclusion

Security module in web environment was designed for safe data sending. To provide abilities of confidentiality, integrity, digital signature, we use PKI public key infrastructure and it was designed secure modules to

insert/delete dynamically. Secure module was designed for the server and client. In security module of server, the HTML page was enciphered each character with 512 bits key value to confident data. Also, SHA-1 hash algorithm was used to test data integrity of transmitted data. The encrypting messages from client are decrypted with RSA algorithm in server side, the decrypted message are stored into data base file and acknowledged to client as normal.

Client security module decrypts enciphered messages to normal message using RSA algorithm. Input data of CGI form was enciphered and sent to server system. To classify, URL filtering was used in security web. Circumstances of implementation can reciprocate security module data that connect to security web server from URL filtering information. Unless secure module is, the messages are send off normally through URL filtering information.

## references

[1] Warwick Ford, Michael S. Baum, Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption, Prentice Hall, 2000.
[2] Amrit Tiwana, Web Security, Digital Press, 1999.
[3] Lincoln D. Stein, Web Security: A Step-by-Step Reference Guide, Addison-Wesley, 1999.
[4] Donna Woouteiler, Web Security; A Matter of Trust, O'Reilly & Associates , 1997.
[5] Anup K. Ghosh, "E-Commerce Security : Weak Links, Best Defenses", John Wiley & Sons, 1998.
[6] Mohammed J. Kabir, “Apache Server Bible”, IDG Books Worldwide, 1998.
[7] Peter Wainwright ,"Professional Apache", Wrox Press Inc, 1999.
[8] Bob Quinn, Dave Shute, “Windows Sockets Network Programming”, Addison-Wesley ,1995.
[9] Wei Hua, Jim Ohlund, Barry Butterklee , “Unraveling the Mysteries of Writing a Winsock 2 Layered Service Provider ”http://www.microsoft.com/msj, 1999.
[10] Michael Rosing, “Implementing Elliptic Curve Cryptography”, Manning Publications Company; 1998.
[11] Richard E. Smith ,Internet Cryptography , Addison-Wesley, 1997.
[12] Stephen A. Thomas, SSL and TLS Essentials: Securing the Web , John Wiley & Sons, 2000.
[13] Lincoln Stein, Doug MacEachern, Linda Mui, “Writing Apache Modules with Perl and C: The Apache API and mod_perl”, O'Reilly & Associates, 1999.

**Seung-Ju, Jang** received a B.Sc. degree in Computer Science and Statistics, and M.Sc. degree, and his Ph.D. in Computer Engineering, all from Busan National University, in 1985, 1991, and 1996, respectively. He is a member of IEEE and ACM. He has been an associate Professor in the Department of Computer Engineering at Dongeui University since 1996. He was a member of ETRI(Electronic and Telecommunication Research Institute) in Daejon, Korea, from 1987 to 1996, and developed the National Administration Multiprocessor Minicomputer during those years. His current research interests include fault-tolerant computing systems, distributed systems in the UNIX Operating Systems, multimedia operating systems, security system, and parallel algorithms.